

WEST Search History

[Hide Items](#) [Restore](#) [Clear](#) [Cancel](#)

DATE: Friday, December 16, 2005

<u>Hide?</u>	<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>			
<input type="checkbox"/>	L5	L4 and L3	0
<input type="checkbox"/>	L4	((merg\$ or combin\$ or connect\$ or embed\$) near2 object) same version\$.clm.	45
<input type="checkbox"/>	L3	((updat\$ or upgrad\$) same ancestor).clm.	22
<input type="checkbox"/>	L2	6367077.pn.	2
<input type="checkbox"/>	L1	20020129352	2

END OF SEARCH HISTORY

WEST Search History

[Hide Items](#) [Restore](#) [Clear](#) [Cancel](#)

DATE: Friday, December 16, 2005

Hide?	Set <u>Name</u>	Query	Hit Count
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	10
<input type="checkbox"/>	L19	(L17 or L18) and merg\$	2
<input type="checkbox"/>	L18	L13 and ((updat\$ or upgrad\$) same ancestor)	14
<input type="checkbox"/>	L17	L13 and (((merg\$ or combin\$ or connect\$ or embed\$) near2 object) same version\$)	232
<input type="checkbox"/>	L16	L13 and customiz\$	37
<input type="checkbox"/>	L15	L13 and L7	
		<i>DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	0
<input type="checkbox"/>	L14	L13	
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	417
<input type="checkbox"/>	L13	L12 or L11	47
<input type="checkbox"/>	L12	L9 and (generat\$ near new version)	376
<input type="checkbox"/>	L11	L9 and (creat\$ near new version)	2549
<input type="checkbox"/>	L10	L9 and new version	35355
<input type="checkbox"/>	L9	L8 and (differe\$ near between)	98590
<input type="checkbox"/>	L8	version\$ and (updat\$ or upgrad\$)	
		(717/120 717/121 717/122 717/123 717/168 717/169 717/170 717/171 717/172 717/173).ccls.	2066
<input type="checkbox"/>	L7		
		<i>DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ</i>	1
<input type="checkbox"/>	L6	5809287.pn.	
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	0
<input type="checkbox"/>	L5	L4 and L3	
<input type="checkbox"/>	L4	((merg\$ or combin\$ or connect\$ or embed\$) near2 object) same version\$.clm.	45
<input type="checkbox"/>	L3	((updat\$ or upgrad\$) same ancestor).clm.	22
<input type="checkbox"/>	L2	6367077.pn.	2
<input type="checkbox"/>	L1	20020129352	

END OF SEARCH HISTORY

Hit List

First Hit

Search Results - Record(s) 1 through 10 of 10 returned.

1. Document ID: US 20050262165 A1

L19: Entry 1 of 10

File: PGPB

Nov 24, 2005

PGPUB-DOCUMENT-NUMBER: 20050262165
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20050262165 A1

TITLE: Data storage system

PUBLICATION-DATE: November 24, 2005

INVENTOR-INFORMATION:

NAME

Scott, Tim

Goddard, Denis Michael

CITY

Berkshire

Redwood Shores

STATE

CA

COUNTRY

GB

US

US-CL-CURRENT: 707/203

ABSTRACT:

A data storage system for storing a number of versions of a number of objects, the versions of the objects being arranged in branch groups, wherein at least one version of at least one object defines a branch group, and wherein the versions of the objects are stored in accordance with their branch group, the system comprising an object versions table operable to include information relating to each version of each object and identifying, for each version of each object, a branch group to which the version of the object belongs and an object branches table operable to include information relating to each branch group and identifying, for each branch group, each version of each object that belongs to the branch group.

2. Document ID: US 20050234997 A1

L19: Entry 2 of 10

File: PGPB

Oct 20, 2005

PGPUB-DOCUMENT-NUMBER: 20050234997
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20050234997 A1

TITLE: Byte-level file differencing and updating algorithms

PUBLICATION-DATE: October 20, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Gu, Jinsheng	San Mateo	CA	US
Peng, Luosheng	Santa Clara	CA	US

US-CL-CURRENT: 707/104.1

ABSTRACT:

A method for performing differencing and updating between electronic files is provided. A byte-level file differencing algorithm receives two byte streams corresponding to an original file and a new file. The new file includes updated and revised versions of the original file. The file differencing algorithm determines a longest common sub-string (LCS) between the two byte streams and divides each of the two byte streams into sub-streams. The sub-streams include the LCS along with prefix and suffix sub-streams to the LCS. The file differencing algorithm then recursively determines an LCS and divides each sub-stream until a size of the sub-streams is less than a pre-specified size. Byte-level differences are then identified between each of the corresponding sub-streams. Further, the file differencing algorithm defines a protocol for structuring a delta file by using a set of operation codes and a variable length integer format to eliminate redundant information in the delta file. Using this protocol, the file differencing algorithm generates the delta file including an operation array that codes the identified byte-level differences.

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequence	Attachments	Claims	Drawings	Patent
------	-------	----------	-------	--------	----------------	------	-----------	----------	-------------	--------	----------	--------

3. Document ID: US 20050131870 A1

Jun 16, 2005

L19: Entry 3 of 10

File: PGPB

PGPUB-DOCUMENT-NUMBER: 20050131870
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20050131870 A1

TITLE: Versioning in an integration platform

PUBLICATION-DATE: June 16, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Krishnaswamy, Janaki	Redwood Shores	CA	US
Nainani, Bhagat Vikram	Fremont	CA	US
Nickolayev, Oleg Y.	Redwood Shors	CA	US
Saxena, Vishal	Belmont	CA	US
Stallard, William George	San Francisco	CA	US
Thompson, David Wheeler Bruce	San Francisco	CA	US

US-CL-CURRENT: 707/3

ABSTRACT:

A repository contains multiple versions of an object but only a single version of the object is supplied when a query is made. The single version is automatically selected from among a number of versions that are otherwise returned in response to the query, based on a configuration associated with a workspace in which the query originates. The selected version of the object is then presented in a version resolved view, without exposing any information related to versioning of the object. Specifically, a number of configurations are established, each configuration containing no more than one version of each object in the repository. However, only one configuration is associated with each workspace from which a query can originate. The configuration that is associated with the workspace depends on whether the workspace is to be used for design of the repository or for use of the repository during live operation. Specifically, a single configuration (hereinafter "design time" configuration) is commonly associated with the workspaces of all developers. When the developers decide that a set of objects in the repository is ready for use in live operation, the set of objects is "deployed" by copying the design time configuration to generate a new configuration (hereinafter "run time" configuration) that contains the most current versions of all objects (as present in the design time configuration). Any number of run time configurations can co-exist with each other and with the design time configuration.

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequence	Attachments	Claims	None	None
------	-------	----------	-------	--------	----------------	------	-----------	----------	-------------	--------	------	------

4. Document ID: US 20050010576 A1

Jan 13, 2005

L19: Entry 4 of 10

File: PGPB

PGPUB-DOCUMENT-NUMBER: 20050010576
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20050010576 A1

TITLE: File differencing and updating engines

PUBLICATION-DATE: January 13, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Ren, Liwei	Alviso	CA	US
Gu, Jinsheng	Alviso	CA	US
Lai, David	Fremont	CA	US

US-CL-CURRENT: 707/100

ABSTRACT:

A file differencing and updating system is provided that includes a file differencing component and a file updating component. The file differencing component, or file differencing engine, generates a difference file in a first processor-based or computer system from an original or old version and a new version of an electronic file. The file updating component, or file updating engine, generates a copy of the new file on a second processor-based or computer system using the difference file and the hosted copy of the original file.

5. Document ID: US 20030212712 A1

L19: Entry 5 of 10

File: PGPB

Nov 13, 2003

PGPUB-DOCUMENT-NUMBER: 20030212712
 PGPUB-FILING-TYPE: new
 DOCUMENT-IDENTIFIER: US 20030212712 A1

TITLE: Byte-level file differencing and updating algorithms

PUBLICATION-DATE: November 13, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
Gu, Jinsheng	San Mateo	CA	US
Peng, Luosheng	Santa Clara	CA	US

US-CL-CURRENT: 707/200

ABSTRACT:

A method for performing differencing and updating between electronic files is provided. A byte-level file differencing algorithm receives two byte streams corresponding to an original file and a new file. The new file includes updated and revised versions of the original file. The file differencing algorithm determines a longest common sub-string (LCS) between the two byte streams and divides each of the two byte streams into sub-streams. The sub-streams include the LCS along with prefix and suffix sub-streams to the LCS. The file differencing algorithm then recursively determines an LCS and divides each sub-stream until a size of the sub-streams is less than a pre-specified size. Byte-level differences are then identified between each of the corresponding sub-streams. Further, the file differencing algorithm defines a protocol for structuring a delta file by using a set of operation codes and a variable length integer format to eliminate redundant information in the delta file. Using this protocol, the file differencing algorithm generates the delta file including an operation array that codes the identified byte-level differences.

6. Document ID: US 20030115223 A1

L19: Entry 6 of 10

File: PGPB

Jun 19, 2003

PGPUB-DOCUMENT-NUMBER: 20030115223
 PGPUB-FILING-TYPE: new
 DOCUMENT-IDENTIFIER: US 20030115223 A1

TITLE: Data storage system

PUBLICATION-DATE: June 19, 2003

INVENTOR-INFORMATION:

NAME
Scott, Tim
Goddard, Denis Michael

CITY	STATE	COUNTRY
Berkshire	CA	GB
Redwood Shores		US

US-CL-CURRENT: 707/203

ABSTRACT:

A data storage system for storing a number of versions of a number of objects, the versions of the objects being arranged in branch groups, at least one version of at least one object defines a branch group, and the versions of the objects are stored in accordance with their branch group, comprising: an object branches table including a branch column, an object identity column, a version sequence column and an object data column, wherein data identifying the branch group, the object and the number of versions of the object in the branch group are stored in the branch column, object identity column and the version sequence column respectively, and the versions of the objects in each branch group are stored in the object data column, and wherein the object branches table is arranged such that only one row of the object branches table is provided corresponding to each object.

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequence	Attachments	Claims	Print	Frame

7. Document ID: US 6925467 B2

L19: Entry 7 of 10

File: USPT

Aug 2, 2005

US-PAT-NO: 6925467

DOCUMENT-IDENTIFIER: US 6925467 B2

TITLE: Byte-level file differencing and updating algorithms

DATE-ISSUED: August 2, 2005

INVENTOR-INFORMATION:

NAME
Gu; Jinsheng
Peng; Luosheng

CITY	STATE	ZIP CODE
San Mateo	CA	
Santa Clara	CA	

COUNTRY

US-CL-CURRENT: 707/101; 707/1, 707/203, 707/204, 709/203, 714/38, 715/511, 717/168,
717/170

ABSTRACT:

A method for performing differencing and updating between electronic files is provided. A byte-level file differencing algorithm receives two byte streams corresponding to an original file and a new file. The new file includes updated and revised versions of the original file. The file differencing algorithm determines a longest common sub-string (LCS) between the two byte streams and divides each of the two byte streams into sub-streams. The sub-streams include the LCS along with

prefix and suffix sub-streams to the LCS. The file differencing algorithm then recursively determines an LCS and divides each sub-stream until a size of the sub-streams is less than a pre-specified size. Byte-level differences are then identified between each of the corresponding sub-streams. Further, the file differencing algorithm defines a protocol for structuring a delta file by using a set of operation codes and a variable length integer format to eliminate redundant information in the delta file. Using this protocol, the file differencing algorithm generates the delta file including an operation array that codes the identified byte-level differences.

28 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 20

Full	Title	Citation	Front	Revised	Classification	Date	Reference	Claims	TOC	Drawings
------	-------	----------	-------	---------	----------------	------	-----------	--------	-----	----------

8. Document ID: US 6604236 B1

L19: Entry 8 of 10

File: USPT

Aug 5, 2003

US-PAT-NO: 6604236

DOCUMENT-IDENTIFIER: US 6604236 B1

** See image for Certificate of Correction **

TITLE: System and method for generating file updates for files stored on read-only media

DATE-ISSUED: August 5, 2003

INVENTOR-INFORMATION:

NAME

Draper; Stephen Péter Willis

CITY

Hants

STATE

ZIP CODE

COUNTRY

GB

Collins; Brian James

Surrey

GB

Falls; Patrick Terence

Berkshire

GB

US-CL-CURRENT: 717/170

ABSTRACT:

A method and system of the present invention generates a representation of a new version of an original file system with reference to the original file system and the new version of the file system. Use of data from previous versions of the file system reduces the amount of data to be stored in the delta directory map file, delta modification data block file, and delta look up table generated for the data portions unique to the newest version of an original file system. The inventive process produces delta data block records that identify the location of data portions that may be used to generate the newest version of the file system. The data portions may be located in a file in the original file system, a delta modification data block file in a previous version of the original file system or a delta modification data block file for the newest version of the original file system.

The method of the present invention is performed by generating a basis index table identifying the data content of an original file system, generating a file of

modification data blocks that may be used to modify the data content of the original file system, and generating a delta look up table that identifies the location of the data blocks used to represent the newest version of the original file system. The delta look up table and the file of modification data blocks may be stored for delivery to a computer on which a copy of the original file system is stored. The delta look up table and the file of modification data blocks are then used by the computer system on which a copy of the original file system is stored to provide the data content for a new version of the original file system. This is done in way that appears to provide a single file system containing the new version of the file system. Thus, the method of the present invention may be used to generate data for updating the content of a copy of the original file system without having to generate a copy of every file and data block for the new content of the original file system.

164 Claims, 6 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 7

Full	Title	Citation	Front	Review	Classification	Date	Reference	Claims	TOC	Claims
------	-------	----------	-------	--------	----------------	------	-----------	--------	-----	--------

9. Document ID: US 6460052 B1

Oct 1, 2002

L19: Entry 9 of 10

File: USPT

US-PAT-NO: 6460052

DOCUMENT-IDENTIFIER: US 6460052 B1

TITLE: Method and system for performing fine grain versioning

DATE-ISSUED: October 1, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Thomas; Peter Rex	Winkfield			GB
Bradshaw; David	Camberley			GB
Gitterman; Jay	Palo Alto	CA		
Scott; Timothy Nicholas	Ascot			GB

US-CL-CURRENT: 707/203; 707/1, 707/100, 707/104.1, 707/200, 707/201, 707/8

ABSTRACT:

A method and apparatus are provided for converting a non-versioned schema to a version-enabled schema. The method includes reading a first set of table definitions that belong to the non-versioned schema. A second set of table definitions is generated for the version-enabled schema such that each table definition in the second set of table definitions corresponds to a table definition in the first set of table. Each table definition in the second set of table definitions includes columns that correspond to the columns of the corresponding table definition in the first set of table definitions and one or more additional columns for storing version information. Multiple versions of a particular object are stored within the table. In response to a request from a user to retrieve the particular object, a version of the particular object to present to the user is determined based on a workspace associated with the user. The version of the

particular object is presented to the user without exposing values from the second set of one or more columns to the user.

34 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full	Title	Citation	Front	Review	Classification	Date	Reference	Claims	WOFO	Drawn
------	-------	----------	-------	--------	----------------	------	-----------	--------	------	-------

10. Document ID: US 4558413 A

L19: Entry 10 of 10

File: USPT

Dec 10, 1985

US-PAT-NO: 4558413
DOCUMENT-IDENTIFIER: US 4558413 A

TITLE: Software version management system

DATE-ISSUED: December 10, 1985

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schmidt; Eric E.	Los Altos	CA		
Lampson; Butler W.	Philadelphia	PA		

US-CL-CURRENT: 707/203; 717/110, 717/145, 717/171

ABSTRACT:

A software version management system, also called system modeller, provides for automatically collecting and recompiling updated versions of component software objects comprising a software program for operation on a plurality of personal computers coupled together in a distributed software environment via a local area network. The component software objects include the source and binary files for the software program, which stored in various different local and remote storage means through the environment. The component software objects are periodically updated, via a system editor, by various users at their personal computers and then stored in designated storage means. The management system includes models which are also objects. Each of the models is representative of the source versions of a particular component software object and contain object pointers including a unique name of the object, a unique identifier descriptive of the chronological updating of its current version, information as to an object's dependencies on other objects and a pathname representative of the residence storage means of the object. Means are provided in the system editor to notify the management system when any one of the objects is being edited by a user and the management system is responsive to such notification to track the edited objects and alter their respective models to the current version thereof.

6 Claims, 29 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 24

Terms

(L17 or L18) and merg\$

Documents

10

Display Format:

Previous Page

Next Page

Go to Doc#